EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Robert Kowert (Reg. No. 39,255) on 2/27/2009.

2. In the response filed 8/21/2007, applicant argued that claim 34 be examined under 35 U.S.C. 112 Sixth Paragraph, and in the interview on 2/27/2009, Applicant stated the structure for carry out the functions is described in the specification, page 10, line 21 – page 11, line 14.

3. The drawings filed 6/20/2003 are accepted

4. The application has been amended as follows:

Claim 1 (Currently Amended) A system, comprising:

a processor; and

a memory comprising program instructions, wherein the program instructions are executable by the processor to implement:

a virtual machine;

a default class loader for the virtual machine, configured to:

load classes for code executable within the virtual machine on the system from one or more local locations indicated by a class path of the default class loader;

determine that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path; and

generate an indication that the class is not loaded;

a remote class loader mechanism configured to:

detect the indication that the class is not loaded;

obtain the class from a remote system via a network; and

store the class in a location indicated by the class path of the default class loader on the system;

wherein the remote class loader mechanism is not a subclass of the default class loader, and is configured to perform said detect, said obtain, and said store separate from and transparent to the default class loader, and wherein the default class loader is independent from the remote class loader mechanism; ~~and~~

wherein the default class loader is configured to load the class from the location indicated by the class path, and wherein the default class loader being configured to load the class from the location avoids class conflicts; and

wherein, to load the class from the location indicated by the class path, the default class loader is configured to:

locate the class stored in the location indicated by the class path; and load the class from the location for access by the code.

Claim 2 Canceled.

Claim 18 (Currently Amended) A distributed computing system, comprising:

a master node configured to provide computer-executable code fragments of an application to a plurality of worker nodes on a network, wherein the code fragments are configured to run tasks in parallel on two or more of the plurality of worker nodes to perform a job;

a worker node configured to receive a code fragment from the master peer node;

wherein the worker node comprises a virtual machine and a default class loader for the virtual machine, wherein the default class loader is configured to:

load classes for the code fragment executable within the virtual machine from one or more local locations indicated by a class path of the default class loader;

determine that a class needed to execute the code fragment is not stored in the one or more locations indicated by the class path;

wherein the worker node further comprises a remote class loader configured to:

detect that the class is not loaded;

obtain the class from a remote node via the network; and

store the class in a location indicated by the class path of the default class loader on the worker node;

wherein the remote class loader is not a subclass of the default class loader, and is configured to perform said detect, said obtain, and said store separate from and transparent to the default class loader, and wherein the default class loader is independent from the remote class loader; ~~and~~

wherein the default class loader is further configured to load the class from the location indicated by the class path, and wherein the default class loader being configured to load the class from the location avoids class conflicts; and

wherein, to load the class from the location indicated by the class path, the default class loader is configured to:

locate the class stored in the location indicated by the class path; and

load the class from the location for access by the code fragment.

Claim 19 Canceled

Claim 34 (Currently Amended) A system, comprising:

a default class loader means for a virtual machine, wherein the default class loader means is configured to load classes for code executable within the virtual machine on the system from one or more local locations indicated by a class path of the default class loader means;

a remote class loader means comprising:

means for determining that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path;

means for obtaining the class from a remote system via a network; and

means for storing the class in a location on the system indicated by the class path of the default class loader means;

wherein said remote class loader means ~~for determining, said means for obtaining, and said means for storing are~~ is not a subclass of the default class loader means and is configured to operate separate from and transparent to the default class loader, and wherein the default class loader means is independent from said remote class loader means ~~for determining, said means for obtaining, and said means for storing~~; ~~and~~

wherein the default class loader means is configured to load the class from the location indicated by the class path, and wherein the default class loader means being configured to load the class from the location avoids class conflicts; and

wherein, to load the class from the location indicated by the class path, the default class loader means is configured to:

locate the class stored in the location indicated by the class path; and

load the class from the location for access by the code.

Claim 35 (Currently Amended) A method, comprising:

loading classes for code executing within a virtual machine on a system from one or more local locations indicated by a class path of a default class loader for the virtual machine;

determining that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path;

generating an indication that the class is not loaded;

detecting the indication that the class is not loaded;

obtaining the class from a remote system via a network in response to said detecting;

storing the class in a location indicated by the class path of the default class loader on the system;

wherein said loading, said determining, and said generating are performed by the default class loader;

wherein said detecting, said obtaining, and said storing are performed by a remote class loader mechanism, said remote class loader mechanism is not a subclass of the default class loader and is configured to perform said detecting, said obtaining, and said storing separate from and transparent to the default class loader, and wherein the default class loader is independent from said remote class loader mechanism ~~detecting, said obtaining, and said storing~~; ~~and~~

the default class loader loading the class from the location indicated by the class path, wherein the default class loading the class from the location avoids class conflicts; and

wherein said loading the class from the location indicated by the class path further comprises:

the default class loader locating the class stored in the location indicated by the class path; and the default class loader loading the class from the location for access by the code.

Claim 36 Canceled.

Claim 52 (Currently Amended) A computer-accessible storage medium, comprising program instructions, wherein the program instructions are computer-executable to implement:

loading classes for code executing within a virtual machine on a system from one or more local locations indicated by a class path of a default class loader for the virtual machine;

determining that a class needed to execute the code on the system is not stored in the one or more locations indicated by the class path;

generating an indication that the class is not loaded;

detecting the indication that the class is not loaded;

obtaining the class from a remote system via a network;

storing the class in a location indicated by the class path of the default class loader on the system;

wherein said loading, said determining, and said generating are performed by the default class loader;

wherein said detecting, said obtaining, and said storing are performed by a remote class loader mechanism, said remote class loader mechanism is not a subclass of the default class loader and is configured to perform said detecting, said obtaining, and said storing separate from and transparent to the default class loader, and wherein the default class loader is independent from said remote class loader mechanism ~~detecting, said obtaining, and said storing~~; ~~and~~

the default class loader loading the class from the location indicated by the class path, and

wherein the default class loader loading the class from the location avoids class conflicts; and

wherein, in said loading the class from the location indicated by the class path, the program instructions are further computer-executable to implement:

the default class loader locating the class stored in the location indicated by the class path; and

the default class loader loading the class from the location for access by the code.

Claim 53 Canceled.

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to DIEM K. CAO whose telephone number is (571)272-3760. The examiner can normally be reached on Monday - Friday, 7:30AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571) 272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Meng-Ai An/
Supervisory Patent Examiner, Art Unit 2195

DC
March 1, 2009